A Novel Gestural Input Device for Virtual Reality

Christoph Maggioni
SIEMENS AG
Central Research, ZFE ST SN6
Otto Hahn Ring 6, München, Germany
E-Mail: chm@zfe.siemens.de

1 Introduction

Three-dimensional computer applications and user interfaces offer a new qualitative step in man-machine interfaces by relating to the human's natural familiarity to live in a three-dimensional world. 3-D representations and animations of real environments form a new means of communication between man and machine which makes explicit use of the motoric skills humans have learnt and experienced during their whole lives in moving themselves and manipulating objects in their environment. On the other hand computer applications get more and more complex, and their use demands increasingly skilled training. Making computer applications easier to use is probably one of the main future issues in man-machine communication.

Investigating human interactions soon reveals the important role of gestures and the use of three-dimensional space [Fol87], [FEL92]. These may include head movements, eye movements, movements of the whole body or of parts. Our first step towards a natural gesture interface to the computer is to use hand gestures to perform actions in three-dimensional applications. Developing three-dimensional applications thus involves two things, constructing a virtual workspace on one hand, and the gesture recognition interface on the other.

A well-known system for using human hand gestures as a computer interface is the DataGlove [Zim87], a device that consists of a glove with integrated fiber-optic cables for detection of finger movement and a Polhemus sensor for detection of the hand position in space. However, because of the physical layout the DataGlove restricts the user in his normal way of interacting with computers. The user cannot type or move around in the room without taking off the glove and is always in fear of damaging the expensive hardware.

Previous approaches to using a computer vision system as an input device for computer applications include the Videodesk [Kru83], the Mandala system [Vin91] and GEST [Seg92]. All these systems have the limitations of dealing with only two-dimensional gestures and require a controlled uniform image background.

In this work, we will present a novel system for navigating and acting in three-dimensional virtual environments by using hand gestures [Wir93]. The system consists of two functional parts, namely the virtual reality system for building the virtual workspace, and the gesture recognition system, which recognizes human gestures as input commands. The novelty is that recognition of human gestures is accomplished with image processing rather than physical measurements. Our system is able to work in real-time under noisy and changing conditions, and detects the three-dimensional position and orientation of the human hand. Some basic gestures are derived from this data and used to control the virtual environment.

Our goal is not to build ´immersive´ virtual reality system that aims to give its user the best possible simulation of an artificial reality. Such systems use complex devices like head mounted displays and data gloves, but do not allow the user to interact with the ´real´ office world. Our goal is to de-

0-7803-1363-1/93 $3.00 © 1993 IEEE

velop systems that benefit from three-dimensional representations of virtual environments but use standard computer monitors and non-obstructive input devices.

2 System View

Figure 1 shows our proposed system structure which involves a classical feedback loop. Image of the user and his hand are grabbed by a CCD-camera. By using image processing techniques the human hand is recognized. Its position and orientation in 3-D space are determined and optionally the bending of the fingers might be calculated as well. Gestures are recognized and the interactions of the user's hand with the application are computed. Finally a three-dimensional model of the virtual hand is rendered on the screen. When moving the real hand, the user sees the corresponding movement of the virtual hand on the screen. Human hand-eye coordination allows the user to control the system even if the image processing is somewhat inaccurate.

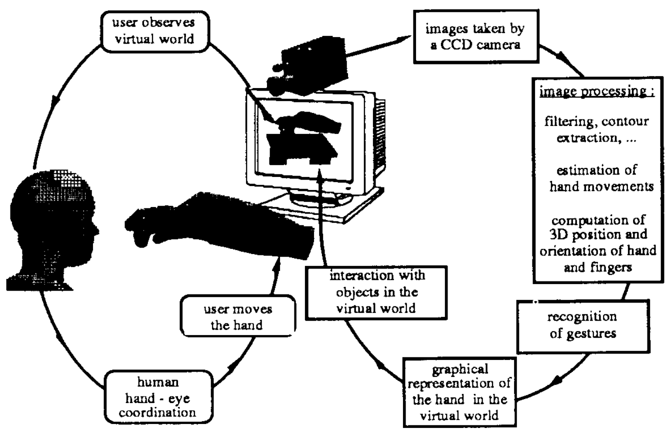

Fig 1: structure of the proposed system

As our system operates in a feedback loop, timing is a crucial issue. If the whole time needed to go through the loop is longer than 1/10 of a second the user will likely get confused and overcompensate, bringing the system to an unstable state.

The image processing system has to satisfy conflicting demands. On one side it has to be reliable, stable and insensitive to changes in the image background or lighting conditions. On the other side it has to be sufficiently fast. As the CCD camera provides an enormous amount of data we have to use some tricks to arrive at a real-time system. We have investigated two main approaches. The first one, which we call ImageGlove, is described in this paper. The approach requires the user to wear a cotton glove with an attached marker to speed up and simplify the image processing task. The other approach, which is not described here, uses the special color of the uncovered human hand as a hint.

The physical layout of the system and the position of the camera depend highly on the application. A natural place to put the camera is on top of the monitor facing the user, where it can also be used for video-conference applications. However, for applications requiring extended use this is not a very good setup. Without support for the elbow, raising the hand all the time tires the arm. (Similar problems are encountered in DataGlove applications.) If we fix the camera above the monitor looking down onto the desk, we can combine the benefits of the mouse with those of our proposed system. 2-D operations allow the user to move his hand on the desktop surface, whereas 3-D gestures can be done by raising the hand. Other possible setups include placing the camera on the ceiling, on a wall or even inside the keyboard looking upwards. All these configurations provide great flexibility to the user.

3 Image Processing Algorithm

One of the most challenging aspects of our system is to extract the necessary information from the images. The camera sends an enormous amount of data (mostly irrelevant) that has to be processed. In order to track the hand, the main task is to extract the translational and rotational parameters in real time. As a first approach we use a marker designed specifically for the task. The marker simplifies the image processing and allows us to build a very fast and reliable system running on a standard workstation. The marker consists of two circles, the outer one white and the inner one black with different center points (Fig. 2a). We currently fix the marker on the back of the hand on a black cotton glove, but a smaller version, i.e. without fingers would be sufficient as well.

The recognition of the marker is based on knowledge of its geometrical properties. All image contours are extracted from the gray-level image (Fig. 2a) and transformed to a higher level description based on moments and chain codes. These structures are used for locating the marker and computing the translation and rotation in three-dimensional space.

The gray-level image of the hand wearing the glove and the background is first binarized by applying a global threshold (Fig. 2b). The thresholding can mostly be done in real-time by setting the input look-up table of the frame grabber board appropriately.

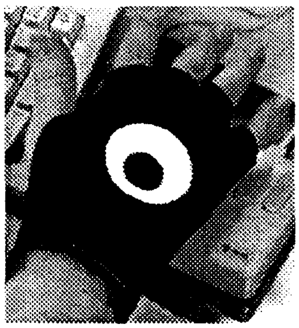

Fig 2a: original image

Fig 2b: binarized image

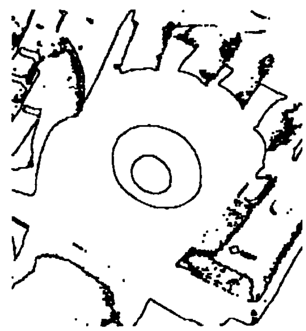

Fig 2c: contours extracted

The contours of the resulting white and black areas are obtained using a contour following algorithm similar to that of Pavlidis (Fig. 2c)[Pav82]. The algorithm scans the image line by line to find a jump in the binary value, follows the contour and stores the chain-code values in a list. As all contours are closed, the process will eventually reach the starting point and scanning of the image continues. By using knowledge about the minimum marker size, we can greatly speed up the scanning process by only looking at every n-th line.

Let the z-axis of the spacial coordinate system point towards the camera and the image plane be parallel to the x-y plane (Fig 3). To estimate all six positional parameters of the marker we first identify the marker in the image, calculate the translational parameters x,y,z and at least estimate the rotational parameters rot_x, rot_y and rot_z.

The recognition of objects independent of their position, size and orientation is an important goal in image processing and methods have been developed that are based on moment invariants [Hu62],[Li91]. The two-dimensional moments of order p,q of an area A computed in a discrete, binarized image with image coordinates x,y is defined as

$$m_{p,q} = \sum_{(x,y)\in A}\sum x^p y^q$$

In specific $m_{0,0}$ is the area of A and $(m_{1,0}/m_{0,0}, m_{1,0}/m_{0,0})$ is the center of mass. The number of operations needed to compute the moments straightforwardley is proportional to the number of

points in A. However, Green's theorem allows us to reduce the amount of calculations by an order of magnitude by just following the border of the area [Li91]. Let A be a simple connected domain and B is the boundary of A in the clockwise direction, the moment calculation is equivalent to

$$m_{p,q} = \frac{1}{p+1} \sum_{(x_i,y_i)\in B}^{N} x_i^{p+1} y_i^q \,\Delta y_i \qquad \text{where } \Delta y_i = \begin{cases} 1 & \text{when } y_{i+1} > y_i \\ 0 & \text{when } y_{i+1} = y_i \\ -1 & \text{when } y_{i+1} < y_i \end{cases}$$

We use the outline of each area computed in the contour following step to calculate the first three moments ($m_{0,0}, m_{1,0}, m_{0,1}$) as well as some other object features. These features are used to search for the marker, which is characterized by the following :

- There are two objects that are circular with approximately the same centers.
- The outer circle is white and the inner one is black.
- The circle size has to be in known upper and lower bounds.
- We known the ratio of circle sizes.
- We know the previous position of the marker in the image.

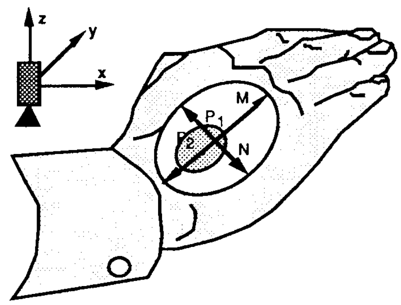

Fig 3: geometrical properties of the marker

The previous constraints are strong and misclassifications are very unlikely. Having detected the marker, we compute the parameters of position and orientation in space. The position in the x and y direction is given by the center of mass derived from the moment calculation. The distance from the camera is a function of the known real marker size, camera parameters such as focal length and the observed area $m_{0,0}$ of the marker in the image.

The rotational parameters can be derived as follows. The distance between the centers of the two circles P_1 and P_2 of the marker gives us the rotation of the hand along the z-axes :

$$rot_z = \arctan\left(\frac{x_2 - x_1}{y_2 - y_1}\right)$$

Due to rotation around the x and y axes, a marker circle in general appears as an ellipse (Fig. 3). The major and minor axes M, N of an ellipse can be computed from the central moments $\tilde{m}_{i,j}$ [Zho89] :

$$\tilde{m}_{i,j} = \frac{1}{p+1} \sum_{(x_i,y_i)\in B}^{N} (x_i - \tilde{x})^{p+1} (y_i - \tilde{y})_i^q \,\Delta y_i \qquad where\ \tilde{x} = \frac{m_{1,0}}{m_{0,0}}\ ,\ \tilde{y} = \frac{m_{0,1}}{m_{0,0}}$$

$$\mathrm{M} = a + \sqrt{a-b} \ \text{ and } \ \mathrm{N} = a - \sqrt{a-b} \ \text{ with } \ a = \frac{\tilde{m}_{2,0} + \tilde{m}_{0,2}}{2},\ b = \left(\tilde{m}_{2,0}\tilde{m}_{0,2} - \tilde{m}_{0,2}^2\right)$$

The angle between the major axis M of the projected circle and the x-axis is

$$\tan 2\theta = \left[\frac{2\tilde{m}_{1,1}}{\tilde{m}_{2,0} + \tilde{m}_{0,2}}\right]$$

It can be shown that the surface normal (A,B,C) to the plane through the marker in 3D-space can be found as follows:

$$A=-B\tan\theta \ , \quad B=\sqrt{1+(\tan\theta)^2+\left(\left(\frac{M}{N}\right)^2-1\right)\frac{1}{(-\tan\theta\sin\theta-\cos\theta)^2}} \ , \quad C=\sqrt{1-A^2-B^2}$$

We want to rotate a computer model of the human hand to fit the observed marker orientation. As rotations in 3D-space are not commutative we have to apply them in a special, predefined order. We choose the order y-x-z. To accomplish that task we find the inverse transformation around the z-axis to bring the marker into the normal position. As shown earlier the rotation around the z-axis can easily be computed. Values A, B and C are calculated and the remaining rotations are given as :

$$A'=A\cos(rot_z)-B\sin(rot_z) \ , \quad B'=A\sin(rot_z)+B\cos(rot_z) \ ,$$
$$rot_x=\arccos\left(\frac{C}{\sqrt{B'^2+C^2}}\right) , \quad rot_y=\arcsin(-A')$$

Note that we have ambiguities in the sign of the rotation around the x and y axes. In practice we can neglect this problem and constrain the hand rotations only to positive values.

As we are processing real-time image sequences we can apply the knowledge from the current frame in order to predict features of consecutive frames. We use motion-estimation techniques to estimate the size and position of a region of interest to search for the marker in the next image. In practical experiments this reduces the amount of computations by a factor of five.

The knowledge of the number of black and white pixels in the marker pattern is used to calculate a updated threshold for the initial binarization step from the observed gray-values of the image. This is necessary because of changing overall lighting conditions and of local effects like shadowing the palm by moving the hand away from the light sources.

The above algorithm is implemented on a SUN SPARC I with a frame grabber board. This is used for grabbing 512x512x8bit images and piping them through a look-up table for thresholding. All the other computations are done on the host computer itself. For typical images with many contours in the background we get an initial frame-rate of 9 frames/sec, but after locating the marker we can track it at the maximum European frame rate of 25 frames/sec.

The tracking system has proven to be very reliable and fast even in difficult environments with a lot of moving objects in the background and changing light conditions. We can extract all six degrees of translation and rotation of the human hand with enough accuracy to use the ImageGlove system as an input device for a number of three-dimensional applications.

4 Gesture Recognition

We are currently using the ImageGlove system to navigate through three-dimensional worlds and to manipulate three-dimensional objects. The observer is modeled by a virtual body whose position and rotation determines his point of view, and a virtual hand that is used to manipulate objects.

The goal is to find an easy way to control the virtual body and hand with the image glove. To accomplish that we have developed two approaches. The first one is used in the system setup where the camera faces the user.

We found this system setup to have one major drawback: raising the hand all the time tires the arm. We therefore developed a new configuration (Fig. 4) where the camera is mounted above the table monitoring a certain volume in space to the right or left of the computer screen.

When the user is performing tasks, requiring 2-D input, the hand can be moved on the tabletop like in a conventional mouse interface or Krueger's system [Kru83]. In this mode the visual feedback for the user is just a 2D drawing of a hand. When the user raises the hand to make a complex gesture or to move objects in 3D space the 3-D mode is automatically entered. As long as the real hand is in the inner part of the control volume it controls the movement of the virtual hand along all

six axes, but when the hand enters the border region the observers viewpoint moves in the corresponding direction (Fig. 4). To provide feedback to the user concerning this mode switch the color of the virtual hand is changed.

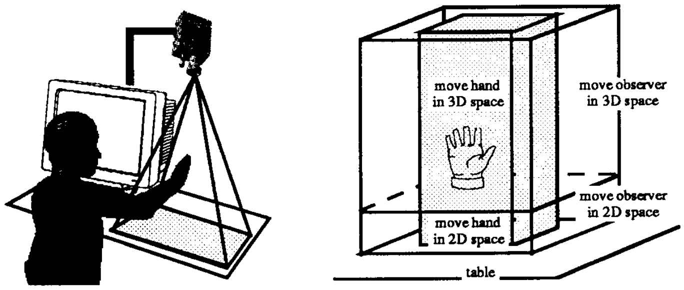

Fig. 4 : Control volume for the hand input

Rotating the viewpoint is done incrementally by rotating the palm around the horizontal axes. Tipping the hand allows one to grab objects. Touching a virtual object triggers actions like information retrieval or changes in internal state.

We have found the described user interface to be very useful in moving around and grabbing objects in a virtual world and to be much more ergonomic in most situations than the one with the camera facing the user.

5 Using the System in Virtual Environments

One powerful application of the ImageGlove is in the area of visualization and virtual reality environments. To test the usefulness of our input device, the reliability of its algorithms and to demonstrate possible applications, we have developed a virtual reality toolkit. It is built using a system called DIVE (Distributed-Virtual-Environment) developed by the Swedish Institute of Computer Science [Car92] that itself relies on ISIS, a distributed message system by Cornell University. The modular design of our software allows us to easily attach different versions of the image processing system, to try out different modules for gesture recognition or to use other three dimensional input devices for comparison. Each virtual world can be shared by an arbitrary number of users and applications that modify objects in the world.

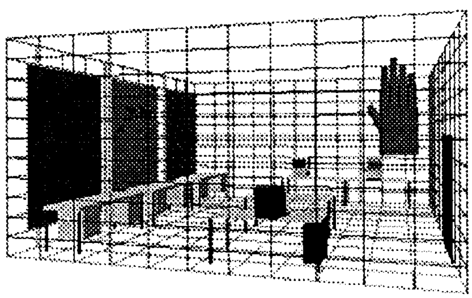

Fig. 5a: SIEMENS Lab

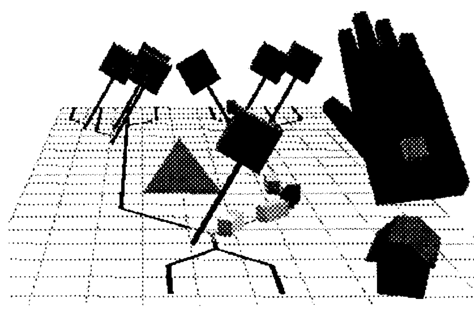

Fig. 5b: Marshaling-Yard Example

We have designed several virtual environments to illustrate possible applications. One is a simulation of our lab at SIEMENS, Munich, in which one can fly around and move objects (Fig. 5a). In another example a marshaling-yard with animated trains is simulated(Fig. 5b). The user can fly around, touch objects like cars and points and thus get information about them, or switch the

points by touching them with a special gesture and thereby cause the trains to change tracks. The virtual reality system has been proven to be flexible and modular. The only problem we have to deal with is the relative slow refresh rate of the 3D-rendering system of 15-10 frames/sec, far below the speed of our image processing system with 25 frames/sec.

6 Conclusion

The ImageGlove is a novel 3-D input device with six-degrees of freedom that can be used to interface with non immersive virtual environments. We are able to detect the position and orientation of the human hand in real time and therefore our device can be used as a wireless replacement for the polhemus position sensor. The system allows one to use the human hand as an input device to interact with three dimensional virtual-reality applications in a very natural way.

In our experience, the camera-based device has proven to be more natural than other devices we have tested such as the spaceball and the mouse. This is especially true in tasks that require unskilled users to perform complex three dimensional operations.

Current work focuses on new image processing techniques for detecting the human hand without using a marker. We are also extending the image processing to incorporate the detection of the fingertips. This will allow us to detect more complex gestures that are not based on special movements of the human hand but rather on the position of the fingertips.

Acknowledgment

I want to thank all members of the GestureComputer project : Subutai Ahmad, Daniel Goryn, Rolf Schuster and Brigitte Wirtz and Søren Hein for proofreading.

References

[Bol87] R.A.Bolt, The integrated multi-modal interface, Trans. IEICE, J70-D:2017-2025,1987

[Car92] C. Carlson, O. Hagsand, The MultiG Distributed Interactive Virtual Enironment, Proceedings of the 5th MultiG Workshop, Stockholm, 1992

[Cla91] M.A. Clarkson, An easier interface, BYTE 16(2), February 1991

[Fel92] W. Felger, How interactive visualization can benefit from multidimensional input devices, Visual Data Interpretation, Proc. SPIE 1668, 1992

[Fol87] J.D. Foley, Interfaces for advanced computing, Scientific American, 257:7, 1987

[Hu62] M.K. Hu, Visual pattern recognition by moment invariants, IRE Trans. Inform. Theory IT-8, 1962, 179-187

[Li91] Bing-Cheng Li and Jun Shen, Fast Computation of Moment Invariants, Pattern Recognition Vol 24, No. 8, 807-813, 1991

[Kru83] M.W. Krueger, "Artificial Reality", Addison-Wesley, 1983

[Pav82] T. Pavlidis: Algorithms for Graphics and Image Processing, Springer 1982

[Seg92] J. Segen, Gest: A learning computer vision system that recognizes gestures, to appear in Machine Learning 4

[Vin91] V. J. Vincent, Dwelving in the depth of the mind, Proc. Interface to real & virtual worlds, Montpellier, 1991

[Wir93] B Wirtz, Ch. Maggioni, ImageGlove : A Novel Way to Control Virtual Environments, Proceedings of the Virtual Reality Systems '93, New York

[Zim87] T.G. Zimmermann, J.Lanier, C. Blanchard, S. Bryson, Y. Harvill, A Hand Gesture Interface Device, Proc. ACM CHI+GI Conf. Human Factors in Computing Systems and Graphics Interface, pp. 189-192

[Zho89] Z.Zhou, K.C.Smith, B.Benhabib, R. Safaee-Rad, Morphological Skeleton Transforms for Determining position and orientation of Pre-Marked Objects, IEEE Pacific Rim Conference on Communication, Computers and Signal Processing, pp. 301-305, 1989